

How to develop your first cloud-native Applications with Java

Niklas Heidloff
Developer Advocate, IBM
[@nheidloff](#)

Harald Uebele
Developer Advocate, IBM
[@Harald_U](#)

Once upon a time ...



“Never not be afraid”

Grug Crood
Beginning of 'The Croods'



@nheidloff @Harald_U

#IBMDeveloper github.com/nheidloff/cloud-native-starter

“Microservices are a software development technique [...] that structures an application as a collection of loosely coupled services.”

Wikipedia

What are cloud-native Applications?

Elasticity

→ App stays responsive

Continuous delivery

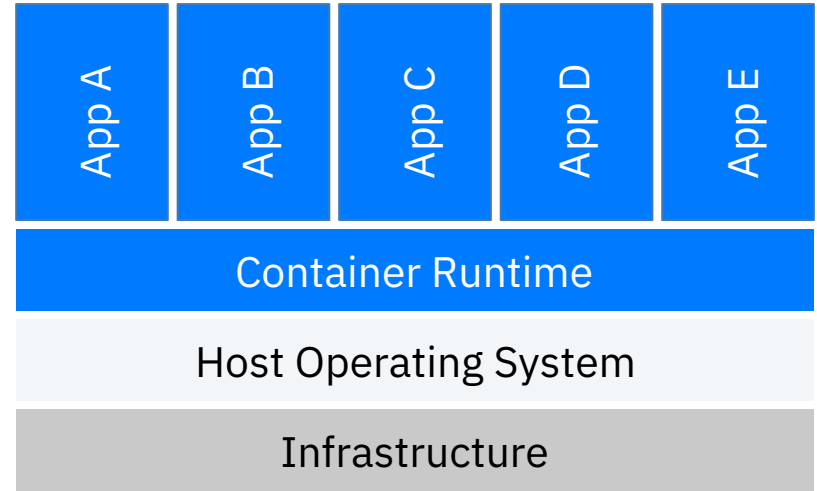
→ DevOps

New Options → New Challenges



“A container image is a lightweight, standalone, executable package of software that includes everything needed to run an application.”

docker.com



“I’m a caveman. Can you explain containers so that I understand it?”

Grug Crood
Caveman learning microservices



@nheidloff @Harald_U

Portable Containers



Java Image

Open source stack

OpenJ9 0.12.1

OpenJDK 8u202-b08 from AdoptOpenJDK

Open Liberty 18.0.0.4

MicroProfile 2.1

@nheidloff @Harald_U

Dockerfile

```
FROM openliberty/open-liberty:microProfile2-java8-openj9
```

```
COPY liberty/server.xml /config/
```

```
ADD target/articles.war /config/dropins/
```

#IBMDeveloper github.com/nheidloff/cloud-native-starter

“Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications.”

kubernetes.io

@nheidloff @Harald_U



kubernetes

#IBMDveloper github.com/nheidloff/cloud-native-starter

“When a container contains everything to run a microservice, why do I need Kubernetes?”

Grug Crood
Caveman learning microservices

@nheidloff @Harald_U







Example Application

Cloud Native Starter

user@demo.email ▾

Articles

 Title	 Author	 Twitter	 Blog
Debugging Microservices running in Kubernetes	Niklas Heidloff	@nheidloff	Blog
Dockerizing Java MicroProfile Applications	Niklas Heidloff	@nheidloff	Blog
Install Istio and Kiali on IBM Cloud or Minikube	Harald Uebele	@harald_u	Blog
Three awesome TensorFlow.js Models for Visual Recognition	Niklas Heidloff	@nheidloff	Blog
Blue Cloud Mirror Architecture Diagrams	Niklas Heidloff	@nheidloff	Blog

Example Application – REST APIs

Browser

Kubernetes with Istio



Web-App



Web-API



Articles



Authors

Exposing REST APIs

JAX-RS

Java API for RESTful Web Services

@nheidloff @Harald_U

GetArticles.java

```
@RequestScoped
@Path("/v1")
@OpenAPIDefinition(info = @Info(title = "Web-API Service",
    version = "1.0", description = "Web-API Service APIs"))
public class GetArticles {

    @Inject
    com.ibm.webapi.business.Service service;

    @GET
    @Path("/getmultiple")
    @Produces(MediaType.APPLICATION_JSON)
    @APIResponses(value = {
        @APIResponse(responseCode = "200",
            description = "Get most recently added articles",
            content = @Content(mediaType = "application/json",
                schema = @Schema(type = SchemaType.ARRAY,
                    implementation = Article.class))),
        @APIResponse(responseCode = "500",
            description = "Internal service error") })
    @Operation(summary = "Get most recently added articles",
        description = "Get most recently added articles")
    public Response getArticles() {
```

Exposing REST APIs

Open API (formerly Swagger)

API description format for REST APIs

@nheidloff @Harald_U

GetArticles.java

```
@RequestScoped
@Path("/v1")
@OpenAPIDefinition(info = @Info(title = "Web-API Service",
    version = "1.0", description = "Web-API Service APIs"))
public class GetArticles {

    @Inject
    com.ibm.webapi.business.Service service;

    @GET
    @Path("/getmultiple")
    @Produces(MediaType.APPLICATION_JSON)
    @APIResponses(value = {
        @APIResponse(responseCode = "200",
            description = "Get most recently added articles",
            content = @Content(mediaType = "application/json",
                schema = @Schema(type = SchemaType.ARRAY,
                    implementation = Article.class))),
        @APIResponse(responseCode = "500",
            description = "Internal service error") })
    @Operation(summary = "Get most recently added articles",
        description = "Get most recently added articles")
    public Response getArticles() {
```

Exposing REST APIs

Open API (formerly Swagger)

API description format for REST APIs

@nheidloff @Harald_U

The screenshot displays the Open Liberty Web-API Service interface. At the top, the Open Liberty logo is visible. The main heading is "Web-API Service" with version "1.9" and "OAS3" indicators. Below this, the text "Web-API Service APIs" is shown. A "Server" dropdown menu is set to "http://192.168.99.100:31380/web-api". The selected API is "default". The interface shows a list of endpoints, with the selected one being a POST endpoint at "/v1/create" with the description "Create a new article". Below this, a GET endpoint at "/v1/getmultiple" is shown with the description "Get most recently added articles". The "Parameters" section for the selected endpoint is empty, with a "Cancel" button. The "Responses" section shows the "Curl" command: `curl -X GET "http://192.168.99.100:31380/web-api/v1/getmultiple" -H "accept: application/json"`. The "Request URL" is `http://192.168.99.100:31380/web-api/v1/getmultiple`. The "Server response" section shows a "Code" of 200 and a "Response body" containing a JSON object:

```
{
  "id": "1555951929394",
  "title": "Example Java App running in the Cloud via Kubernetes",
  "url": "http://heidloff.net/article/example-java-app-cloud-kubernetes",
  "authorName": "Niklas Heidloff",
  "authorBlog": "http://heidloff.net",
  "authorTwitter": "@nheidloff"
}
```

Consuming REST APIs

MicroProfile Rest Client

Type-safe approach to invoke RESTful services

@nheidloff @Harald_U

AuthorsService.java

```
@RegisterProvider(ExceptionMapperArticles.class)
public interface AuthorsService {

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public Author getAuthor(String name) throws NonexistentAuthor;
}
```

AuthorsServiceDataAccess.java

```
public class AuthorsServiceDataAccess implements AuthorsDataAccess {
    public AuthorsServiceDataAccess() {}

    static final String BASE_URL = "http://authors:3000/api/v1/";

    public Author getAuthor(String name) throws NoConnectivity, NonexistentAuthor {
        try {
            name = URLEncoder.encode(name, "UTF-8").replace("+", "%20");
            URL apiUrl = new URL(BASE_URL + "getauthor?name=" + name);
            AuthorsService customRestClient;
            customRestClient = RestClientBuilder.newBuilder().baseUrl(apiUrl)
                .register(ExceptionMapperAuthors.class).build(AuthorsService.class);

            Author output = customRestClient.getAuthor(name);
            return output;
        } catch (NonexistentAuthor e) {
            e.printStackTrace();
            throw new NonexistentAuthor(e);
        } catch (Exception e) {
            throw new NoConnectivity(e);
        }
    }
}
```


Consuming REST APIs

MicroProfile Rest Client

Type-safe approach to invoke RESTful services

@nheidloff @Harald_U

AuthorsService.java

```
@RegisterProvider(ExceptionMapperArticles.class)
public interface AuthorsService {

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public Author getAuthor(String name) throws NonexistentAuthor;
}
```

AuthorsServiceDataAccess.java

```
public class AuthorsServiceDataAccess implements AuthorsDataAccess {
    public AuthorsServiceDataAccess() {}

    static final String BASE_URL = "http://authors:3000/api/v1/";

    public Author getAuthor(String name) throws NoConnectivity, NonexistentAuthor {
        try {
            name = URLEncoder.encode(name, "UTF-8").replace("+", "%20");
            URL apiUrl = new URL(BASE_URL + "getauthor?name=" + name);
            AuthorsService customRestClient;
            customRestClient = RestClientBuilder.newBuilder().baseUrl(apiUrl)
                .register(ExceptionMapperAuthors.class).build(AuthorsService.class);

            Author output = customRestClient.getAuthor(name);
            return output;
        } catch (NonexistentAuthor e) {
            e.printStackTrace();
            throw new NonexistentAuthor(e);
        } catch (Exception e) {
            throw new NoConnectivity(e);
        }
    }
}
```

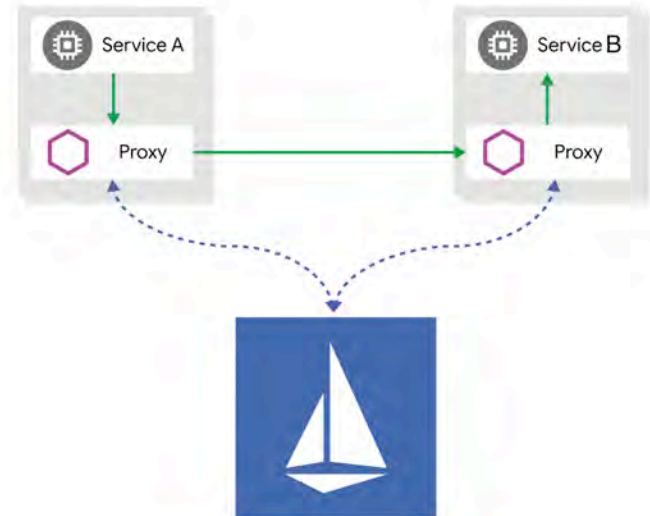
“Istio is an open platform for providing a uniform way to integrate microservices, manage traffic flow across microservices, enforce policies and aggregate telemetry data.”

github.com/istio/istio

@nheidloff @Harald_U



Istio



“Why do I need a service mesh?
Can't I just use Kubernetes?”

Grug Crood
Caveman learning microservices

@nheidloff @Harald_U



Example Application – Traffic Management

Browser

Kubernetes with Istio



Web-App



Web-API v1



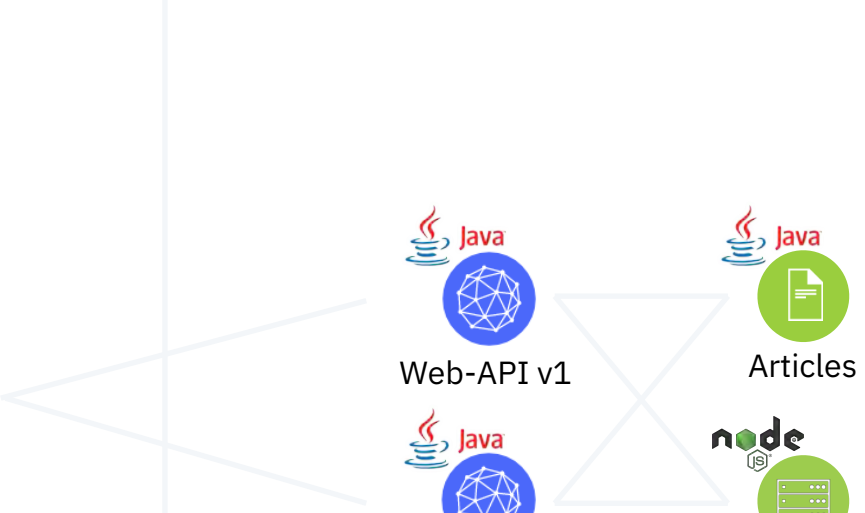
Web-API v2



Articles



Authors



Traffic Management

80% / 20% splitting

@nheidloff @Harald_U

ingress.yaml

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: default-gateway-ingress-http
spec:
  selector:
    istio: ingressgateway
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
    - "*"
---
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
spec:
  hosts:
  - "*"
  gateways:
  - default-gateway-ingress-http
  http:
  - match:
    - uri:
        prefix: /web-api/v1/getmultiple
      route:
        - destination:
            host: web-api
            subset: v1
            weight: 80
          - destination:
            host: web-api
            subset: v2
            weight: 20
```


Traffic Management

Kiali Namespace: default

Graph ? Apr 11, 11:25:29 ... Apr 11, 11:26:29

Display ▾ Edge Labels ▾ Graph Type Versioned app ▾ Find... × Hide... × ?

Fetching Last min ▾ Every 5 sec ▾ ↻

Namespace: default
[applications, services, workloads](#)

Current Graph:
6 apps
4 services
11 edges

HTTP Traffic (requests per second):

Total	%Success	%Error
6.65	100.00	0.00

0 25 50 75 100 %
OK 3xx 4xx 5xx

Service Graph:

- istio-ingressgateway (istio-system) → web-api (85.7%)
- istio-ingressgateway (istio-system) → web-app (14.3%)
- web-api (v1) → articles (v1) (80%)
- web-api (v1) → authors (v1) (20%)
- web-api (v2) → articles (v1) (83.3%)
- web-api (v2) → authors (v1) (91.2%)
- articles (v1) → articles (v1) (100%)
- authors (v1) → authors (v1) (100%)
- web-app (v1) → web-app (v1) (100%)

“Optimizing Enterprise Java for a Microservices Architecture.

[...] by innovating [...] with a
goal of standardization.”

`microprofile.io`

@nheidloff @Harald_U



#IBMDeveloper github.com/nheidloff/cloud-native-starter

“Dude, I just learned Istio and Kubernetes. Can you show me a simple MicroProfile example?”

Grug Crood
Caveman learning microservices

@nheidloff @Harald_U



Example Application – Resiliency

Browser

Kubernetes with Istio



Web-App



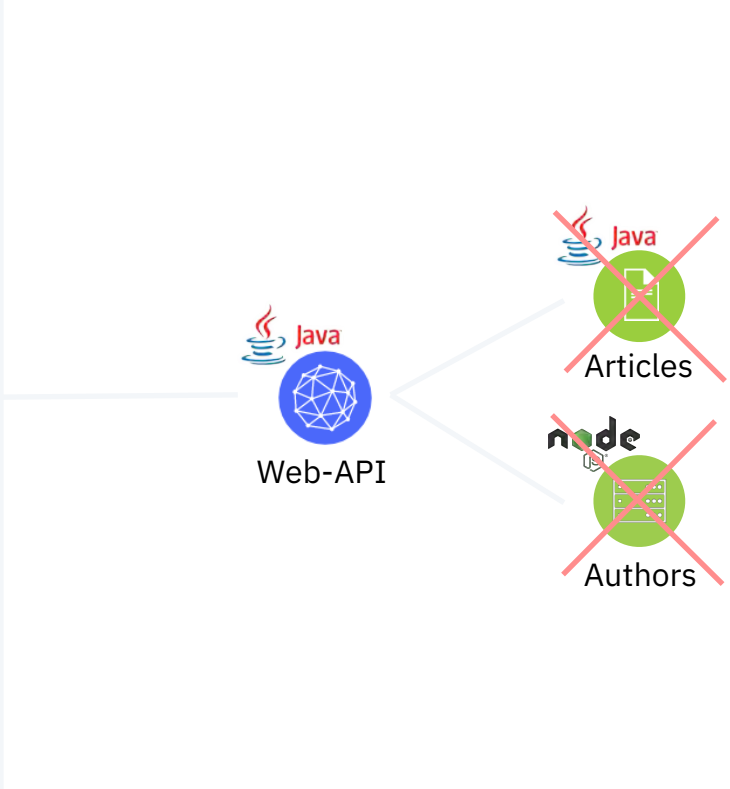
Web-API



Articles



Authors



Resiliency

Authors service not available

Usage of default values

@nheidloff @Harald_U

Service.java

```
private List<Article> lastReadArticles;

public List<Article> fallbackNoArticlesService() {
    return lastReadArticles;
}

@Fallback(fallbackMethod = "fallbackNoArticlesService")
public List<Article> getArticles() throws NoDataAccess {

    List<Article> articles = new ArrayList<Article>();
    List<CoreArticle> coreArticles = new ArrayList<CoreArticle>();

    try {
        coreArticles = DataAccessManager.getArticlesDataAccess().getArticles(5);
    } catch (NoConnectivity e) {
        throw new NoDataAccess(e);
    }

    for (int index = 0; index < coreArticles.size(); index++) {
        CoreArticle coreArticle = coreArticles.get(index);
        Article article = new Article(coreArticle.id, coreArticle.title,
            coreArticle.title, coreArticle.author);
        try {
            Author author;
            author = DataAccessManager.getAuthorsDataAccess().getAuthor(coreArticle.author);
            article.authorBlog = author.blog;
            article.authorTwitter = author.twitter;
        } catch (Exception e) {
            article.authorBlog = "";
            article.authorTwitter = "";
        }
        articles.add(article);
    }
}
```


Resiliency

Articles service not available

MicroProfile Fallback annotation

Service.java

```
private List<Article> lastReadArticles;
```

```
public List<Article> fallbackNoArticlesService() {  
    return lastReadArticles;  
}
```

```
@Fallback(fallbackMethod = "fallbackNoArticlesService")  
public List<Article> getArticles() throws NoDataAccess {  
  
    List<Article> articles = new ArrayList<Article>();  
    List<CoreArticle> coreArticles = new ArrayList<CoreArticle>();  
  
    try {  
        coreArticles = DataAccessManager.getArticlesDataAccess().getArticles(5);  
    } catch (NoConnectivity e) {  
        throw new NoDataAccess(e);  
    }  
  
    for (int index = 0; index < coreArticles.size(); index++) {  
        CoreArticle coreArticle = coreArticles.get(index);  
        Article article = new Article(coreArticle.id, coreArticle.title,  
            coreArticle.title, coreArticle.author);  
        try {  
            Author author;  
            author = DataAccessManager.getAuthorsDataAccess().getAuthor(coreArticle.author);  
            article.authorBlog = author.blog;  
            article.authorTwitter = author.twitter;  
        } catch (Exception e) {  
            article.authorBlog = "";  
            article.authorTwitter = "";  
        }  
        articles.add(article);  
    }  
}
```

Resiliency

Articles



Title

[Debugging Microservices running in Kubernetes](#)

[Dockerizing Java MicroProfile Applications](#)

[Install Istio and Kiali on IBM Cloud or Minikube](#)

[Three awesome TensorFlow.js Models for Visual Recognition](#)

[Blue Cloud Mirror Architecture Diagrams](#)



Author

Niklas Heidloff

Niklas Heidloff

Harald Uebele

Niklas Heidloff

Niklas Heidloff



Twitter



Blog

Authentication and Authorization

OpenID Connect

Identity layer on top of the OAuth 2.0 protocol

IBM App ID

IBM service to authenticate users and protect APIs

policy.yaml

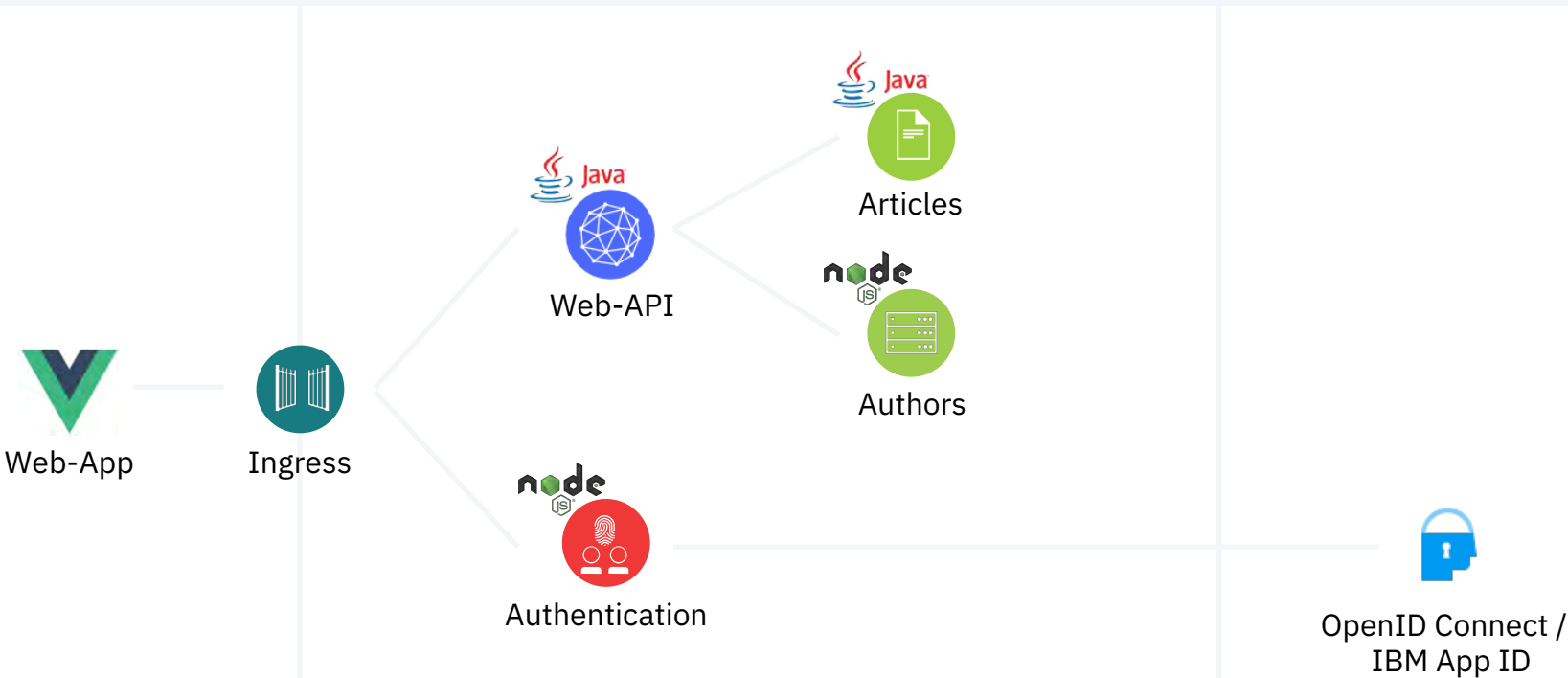
```
apiVersion: "authentication.istio.io/v1alpha1"
kind: "Policy"
metadata:
  name: "protect-web-api"
spec:
  targets:
  - name: web-api
  origins:
  - jwt:
      issuer: "https://us-south.appid.cloud.ibm.com/oauth/v4/xxx"
      jwksUri: "https://us-south.appid.cloud.ibm.com/oauth/v4/xxx/publickeys"
      trigger_rules:
      - included_paths:
        - exact: /web-api/v1/create
  principalBinding: USE_ORIGIN
```

Example Application – Authentication and Authorization

Browser

Kubernetes with Istio

Optional Cloud Services



Authentication



Email:

A light blue input field with a user icon on the left and the text "user@demo.email" inside.

Password:

A light blue input field with a lock icon on the left and seven dots representing a password.

[Forgot Password?](#)

Login

Don't have an account? [Sign up](#)

Powered by App ID

Authorization

Create new Article

Title:

URL:

Author:

Submit

Show Articles

Authorization

Articles

Title	Author	Twitter	Blog
Example Java App running in the Cloud via Kubernetes	Niklas Heidloff	@nheidloff	Blog
Debugging Microservices running in Kubernetes	Niklas Heidloff	@nheidloff	Blog
Dockerizing Java MicroProfile Applications	Niklas Heidloff	@nheidloff	Blog
Install Istio and Kiali on IBM Cloud or Minikube	Harald Uebele	@harald_u	Blog
Three awesome TensorFlow.js Models for Visual Recognition	Niklas Heidloff	@nheidloff	Blog

The screenshot shows the Network tab in a browser's developer tools. The selected request is 'create'. The 'Response Headers' section is expanded, and the 'Request Headers' section is highlighted with a blue box. The 'Request Headers' section shows 'Provisional headers are shown' and includes an 'Authorization' header with the value 'Bearer eyJhbGciOiJIUzI1NiIsInR5cGU6IiwiZW50bnR5bWVudCI6IiJ0IiwiaWF0IjoiYXVhbnR5bWVudCJl'.

```
Filter: Hide data URLs
XHR JS CSS Img Media Font Doc WS Manifest Other
Name: Headers Preview Response Timing
auth
callback?code=w...
loginwithtoken?na...
app.9cd06f5a.css
chunk-vendors.73...
app.bfe2e0c6.js
chunk-vendors.3e...
getmultiple
create
getmultiple

General
Request URL: http://192.168.99.100:31380/web-api/v1/create
Request Method: POST
Status Code: 201 Created
Remote Address: 192.168.99.100:31380
Referrer Policy: no-referrer-when-downgrade

Response Headers
access-control-allow-credentials: true
access-control-allow-headers: origin, content-type, accept, aut
n
access-control-allow-methods: GET, POST, PUT, DELETE, OPTIONS,
access-control-allow-origin: *
content-language: en-US
content-length: 182
content-type: application/json
date: Fri, 12 Apr 2019 06:52:09 GMT
server: istio-envoy
x-envoy-upstream-service-time: 346
x-powered-by: Servlet/4.0

Request Headers
Provisional headers are shown
Accept: application/json, text/plain, */*
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cGU6IiwiZW50bnR5bWVudCI6IiJ0IiwiaWF0IjoiYXVhbnR5bWVudCJl
kLTRmTQZNG2mLWNkOTM1NDMSY11h
Iy0jM1ljSkNCISInZlciI6Ii6NH0.eyJ
3VKLmlibS5jb20vb2F1dGgvdjQvVNG
ZTY01w1YXVhbnR5bWVudCJl
```

Authorization

Via MicroProfile

Manage.java

```
@RequestScoped
@Path("/v1")
public class Manage {

    @Inject
    private JsonWebToken jwtPrincipal;

    @POST
    @Path("/manage")
    @Produces(MediaType.APPLICATION_JSON)
    @Operation(summary = "Manage app", description = "Manage app")
    public Response manage() {
        System.out.println("com.ibm.web-api.apis.Manage.manage");
        System.out.println(this.jwtPrincipal);

        String principalEmail = this.jwtPrincipal.getClaim("email");
        if (principalEmail.equalsIgnoreCase("admin@demo.email")) {
            JsonObject output = Json.createObjectBuilder()
                .add("message", "success").build();
            return Response.ok(output).build();
        }
    }
}
```

“Microservices sound great,
but where is the log file?”

Grug Crood
Caveman learning microservices

@nheidloff @Harald_U



Observability

Tracing
Logging
Monitoring
Metrics
Healthchecks

Microservices vs monolith
→ Higher complexity
→ Ephemeral

Chained invocations

Kubernetes
→ 1 service = N pods

Tracing

OpenTracing

Vendor-neutral APIs and instrumentation for distributed tracing

Jaeger and Zipkin

Open source distributed tracing systems

server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="OpenLiberty Server">

  <featureManager>
    <feature>webProfile-8.0</feature>
    <feature>microProfile-2.1</feature>
    <feature>usr:opentracingZipkin-0.31</feature>
  </featureManager>

  <httpEndpoint id="defaultHttpEndpoint" host="*"
    httpPort="8080" httpsPort="9443"/>

</server>
```

Distributed Tracing

Jaeger UI Lookup by Trace ID... Search Compare Dependencies About Jaeger ▾

← web-api.default: articles.default.svc.cluster.local:8080/* 5366514 Search... Trace Timeline ▾

Trace Start **April 11, 2019 11:04 AM** Duration **14.88ms** Services **2** Depth **2** Total Spans **2**

Service & Operation 0ms 3.72ms 7.44ms 11.16ms 14.88ms

web-api.default articles.default.svc.cluster.local:8080/*

articles.default.svc.cluster.local:8080/* Service: **web-api.default** Duration: **14.88ms** Start Time: **0ms**

- Tags: component=proxy node_id=sidecar-172.17.0.17~web-api-v1-545f655f67-dk59d.default~default.svc.cluster.local guid:x-request-id=0ccde059-534...
- Process: ip=172.17.0.17

SpanID: f1bb534c6a3262ed

articles.default articles.default.svc.cluster.local:8080/* 13.21ms

articles.default.svc.cluster.local:8080/* Service: **articles.default** Duration: **13.21ms** Start Time: **0.33ms**

Tags

- component: "proxy"
- node_id: "sidecar-172.17.0.19-articles-5d6c468d47-fvg8m.default-default.svc.cluster.local"
- guid:x-request-id: "0ccde059-534f-995d-a4ff-849e2deb699e"

Metrics

Prometheus

Monitoring system and time series database

@nheidloff @Harald_U

GetArticles.java

```
@Timed(name = "getArticlesTimed",
    absolute = true,
    displayName = "web-api /getmultiple timer",
    description = "Time taken by getArticles")
@Counted(name = "getArticlesCounted",
    absolute = true,
    displayName = "web-api /getmultiple count",
    description = "Number of times getArticles has been invoked",
    monotonic = true)
@Metered(name = "getArticlesMetered",
    displayName = "web-api /getmultiple frequency",
    description = "Rate the throughput of getArticles")
@GET
@Path("/getmultiple")
@Produces(MediaType.APPLICATION_JSON)
public Response getArticles() {
```

#IBMDeveloper github.com/nheidloff/cloud-native-starter

Metrics

Enable query history

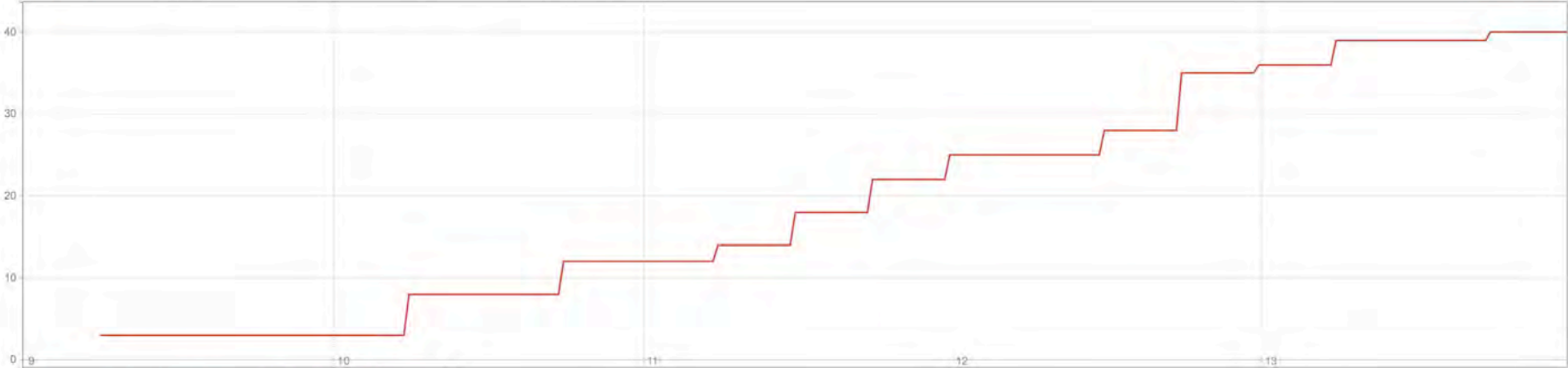
application:get_articles_counted

Execute - insert metric at cursor

Load time: 13ms
Resolution: 1s
Total time series: 1

Graph Console

5m Until Res. (s) stacked



application:get_articles_counted[instance="172.17.0.19:9080",job="webapi"]

Remove Graph

LogDNA

The screenshot displays the LogDNA interface. On the left is a dark sidebar with navigation icons: a red hexagon with a white DNA helix, a lightning bolt, a square with a white circle, a white square with a white circle, a white square with a white circle, a white gear, a white eye, and a white question mark. Below these icons is a search bar with the text 'Find a View'. The main content area is titled 'cloud-native-starter' and contains a log stream. The log entries are as follows:

```
cloud-native-starter ▾ All Tags ▾ All Sources ▾ 3 Apps ▾ All Levels ▾  
Niklas Heidloff  
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.702] [INFO] AuthorsService -  
  {"name":"Niklas Heidloff","twitter":"@heidloff","blog":"http://heidloff.net"}  
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.702] [INFO] AuthorsService -  
  ::ffff:127.0.0.1 - - "GET /api/v1/getauthor?name=Niklas%20Heidloff HTTP/1.1" 200 - "" "Apache-CXF/3.2.6"  
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.711] [INFO] AuthorsService - Query for:  
  Harald Uebele  
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.711] [INFO] AuthorsService -  
  {"name":"Harald Uebele","twitter":"@harald_u","blog":"https://haralduebele.blog"}  
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.712] [INFO] AuthorsService -  
  ::ffff:127.0.0.1 - - "GET /api/v1/getauthor?name=Harald%20Uebele HTTP/1.1" 200 - "" "Apache-CXF/3.2.6"  
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.716] [INFO] AuthorsService - Query for:  
  Niklas Heidloff  
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.720] [INFO] AuthorsService -  
  ::ffff:127.0.0.1 - - "GET /api/v1/getauthor?name=Niklas%20Heidloff HTTP/1.1" 200 - "" "Apache-CXF/3.2.6"  
Apr 8 10:12:14 web-api-v2-5b4d66d87-96m14 web-api com.ibm.web-api.apis.GetArticles.getArticles  
Apr 8 10:12:14 web-api-v2-5b4d66d87-96m14 web-api err [err] com.ibm.webapi.business.getArticles: Cannot connect to  
  articles service  
Apr 8 10:12:14 web-api-v2-5b4d66d87-96m14 web-api err [err] com.ibm.webapi.business.fallbackNoArticlesService:  
  Cannot connect to articles service  
Apr 8 10:12:17 articles-76678b7787-k9rbg articles com.ibm.articles.apis.GetArticles.getArticles  
Apr 8 10:12:17 web-api-v1-567b8cfd4f-8zw58 web-api com.ibm.web-api.apis.GetArticles.getArticles
```

At the bottom of the interface, there is a search bar with the text 'Search...', a 'Jump to timeframe' button, and a 'LIVE' indicator with a red dot.

Dashboards

Add Dashboard

Search dashboards

My Dashboards

Istio 1.0 Overview

Istio 1.0 Service

My Shared Dashboards

HTTP Overview

Network Overview

Overview by Container

Overview by Host

Overview by Process

Top Processes

Dashboards Shared With Me

No dashboards

HTTP Overview

Everywhere



Edit Scope

Request Count

11.4/s

HTTP Error Count

0.21/s

Average Request Time

6.90ms

Max Request Time

106ms

Status Codes Over Time



Average and Max Request Time



Healthchecks

MicroProfile Health

Liveness probes and readiness probes

@nheidloff @Harald_U

HealthEndpoint.java

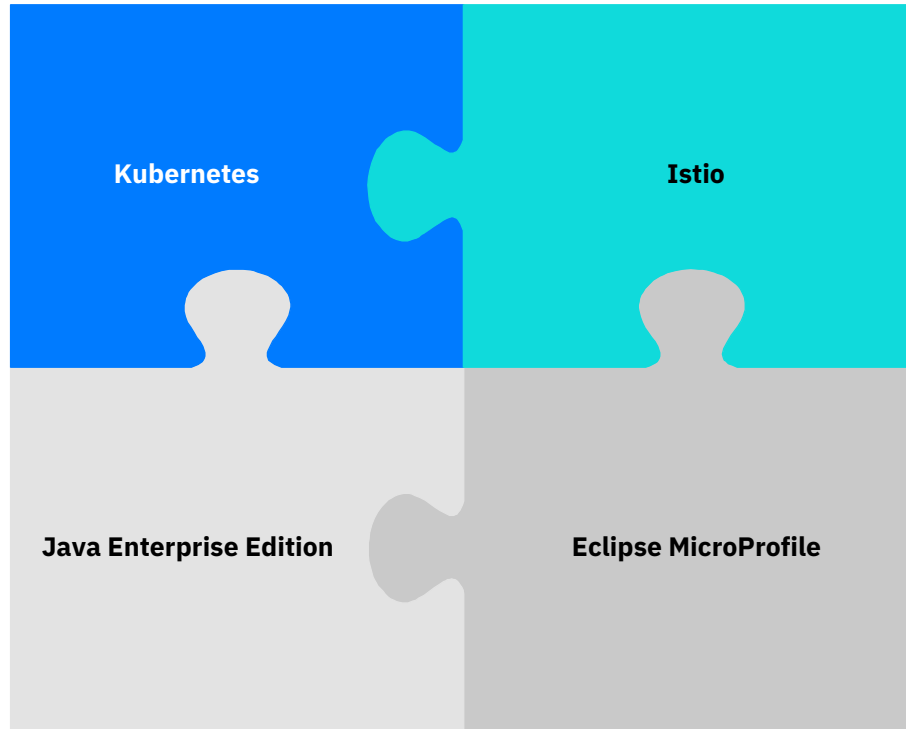
```
@Health
@ApplicationScoped
public class HealthEndpoint implements HealthCheck {

    @Override
    public HealthCheckResponse call() {
        return HealthCheckResponse.named("web-api").withData("web-api", "ok").up().build();
    }
}
```

Service.yaml

```
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: web-api-v1
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: web-api
        version: v1
    spec:
      containers:
        - name: web-api
          image: web-api:1
          ports:
            - containerPort: 9080
          livenessProbe:
            exec:
              command: ["sh", "-c", "curl -s http://localhost:9080/"]
            initialDelaySeconds: 20
          readinessProbe:
            exec:
              command: ["sh", "-c", "curl -s http://localhost:9080/health | grep -q web-api"]
            initialDelaySeconds: 40
          restartPolicy: Always
```

How to use all Pieces together?



Leverage platforms as
much as possible.

Use frameworks for app
specific logic.

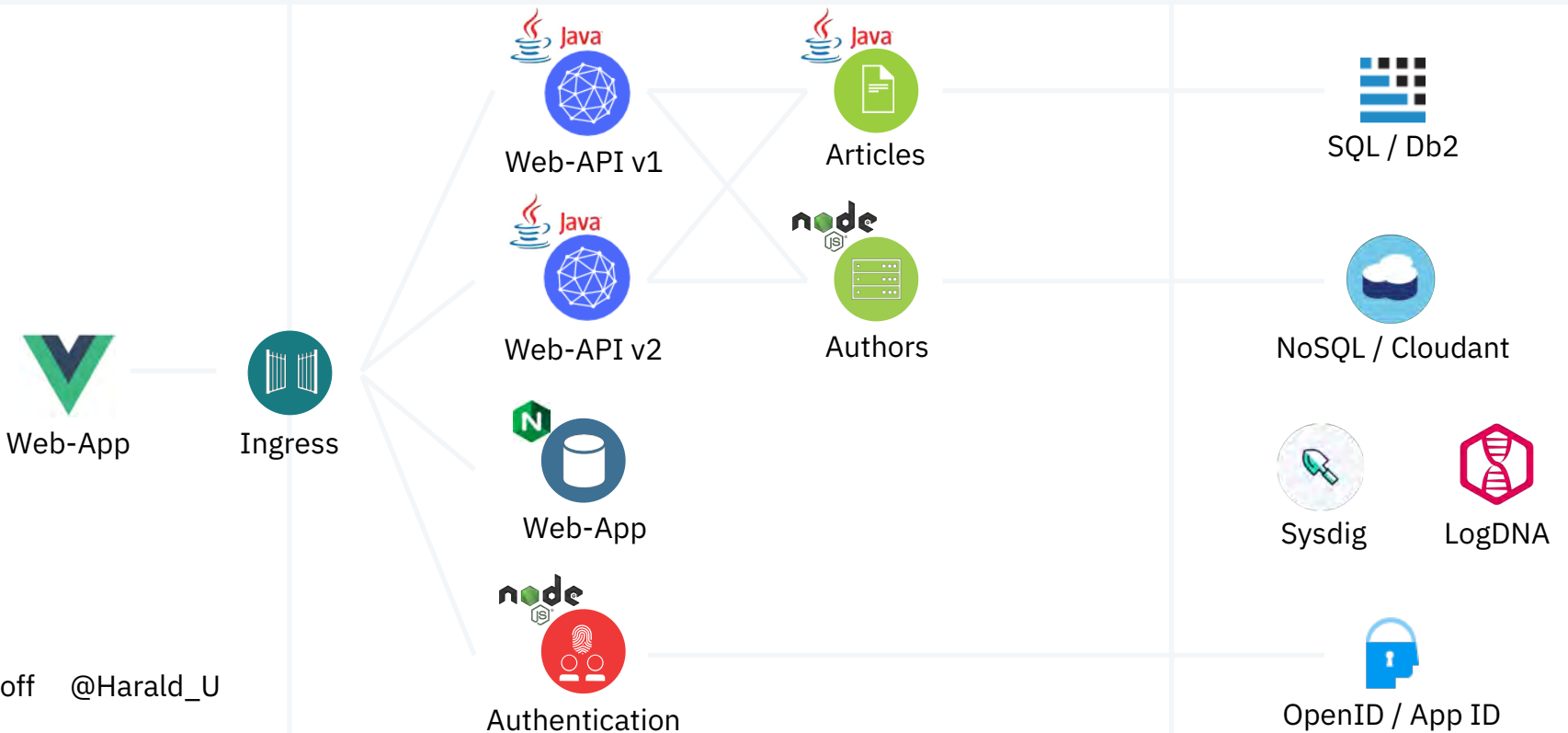
Try out the end-to-end
microservices example
cloud-native-starter!

Architecture: End-to-End Example 'cloud-native-starter'

Browser

Kubernetes with Istio

Optional Cloud Services



IBM Cloud Kubernetes Service including Istio and Knative

The screenshot shows the IBM Cloud Kubernetes Service overview page for a cluster named 'niklas-heidloff-cns'. The cluster is in a 'Normal' state, indicated by a green dot. The page is divided into two main sections: 'Summary' and 'Worker Nodes'. The 'Summary' section lists various cluster details, and the 'Worker Nodes' section shows a progress indicator for 1 node, which is 100% complete and in a 'Normal' state.

IBM Cloud Search resources and offerings... Catalog Docs Support Manage Niklas Heidloff's Account

Clusters / niklas-heidloff-cns

niklas-heidloff-cns ● Normal

Web Terminal (beta) Kubernetes Dashboard Connect via CLI

Access Overview Worker Nodes Worker Pools Add-ons

Summary

Cluster ID	401c8d4144a744f6978c68a12c8335c5
Master Status	Ready
Kubernetes version	1.12.7_1548
Zones	hou02
Owner	niklas_heidloff@de.ibm.com
Resource group	default
Key protect (Beta)	Enable
IAM pullsecrets	Enabled
Public service endpoint URL	https://c5.dal12.containers.cloud.ibm.com:31446 Disable

Worker Nodes 1

100% Normal

1	Normal
0	Warning
0	Critical
0	Pending

“Never not be afraid”

Grug Crood
Beginning of 'The Croods'



@nheidloff @Harald_U

#IBMDeveloper github.com/nheidloff/cloud-native-starter

“Never be afraid”

Grug Crood
End of 'The Croods'



@nheidloff @Harald_U

#IBMDeveloper github.com/nheidloff/cloud-native-starter

Summary

Get the code →



Leverage platforms as much as possible

Use frameworks for app specific logic

IBM loves open source

Kubernetes and Istio

OpenJ9 & AdoptOpenJDK

MicroProfile

Open Liberty

IBM Developer

developer.ibm.com

IBM Cloud Lite account

ibm.biz/nheidloff

@nheidloff @Harald_U

#IBMDveloper github.com/nheidloff/cloud-native-starter

IBM